COBB

Inside MICROSOFT ACCESS™

December 1993 • Vol. 1 No. 10

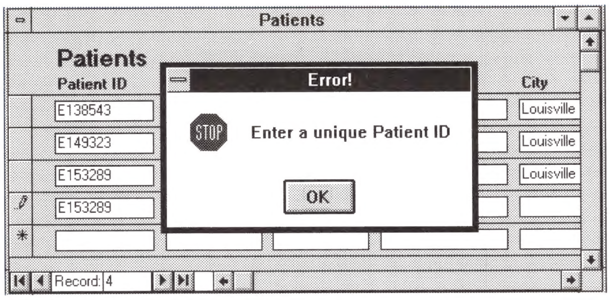
Tips & techniques for Microsoft Access • Windows

Form Tip

Checking for key violations as soon as you leave the primary key field

When you add data to a table, Access checks the entries to ensure they conform to validation rules. If you type an invalid entry, Access immediately beeps and pops up a warning dialog box. You can then provide a valid entry and move on.

Figure A



Enter a unique Patient ID

The macro we'll show you will pop up this message box as soon as you leave the Patient ID field after making a duplicate entry.

It would be nice if Access checked your entries in the primary key field the same way. As you know, you must provide a unique entry for each record's key field. Unfortunately, if you duplicate another record's entry, Access doesn't tell you until you try to post the record.

In this article, we'll show you how to create a macro that will check for key violations as you leave the key field's control—just as Access checks your entries against validation rules. After you implement our technique, a dialog box like the one shown in Figure A will pop up whenever you leave a key field when its entry duplicates another record's entry.

The technique

Let's start with a brief overview of the technique. You first create a macro that actually determines whether you've entered a duplicate key value. You then assign the macro to the Before Update property of the key value's form control. We'll review these two aspects of the technique and then use an example to show you how the technique works.

About the macro

The macro will use the Access Basic function DCount( ) to determine if your entry in the key field already exists. The DCount( ) function determines how many records exist in a table or dynaset that satisfy certain criteria. This sample function call

```
DCount(expr, domain, criteria)
```

shows the function's syntax. The *expr* argument specifies the quantity you want to

IN THIS ISSUE

A Publication of The COBB Group

count, *domain* identifies the table or dynaset that holds the records the function will examine, and *criteria* supplies the criteria a record must meet in order for the function to count the record.

In creating the macro, you'll use DCount( ) to determine how many records exist with the key value you've entered into the form. DCount( ) will return 0 if you've entered a unique value. On the other hand, if another record already uses your entry, DCount( ) will return 1.

If DCount( ) doesn't return 0, the macro will display a message box that warns you about the duplicate key value and will then keep you from moving away from the key field's control. To understand how the macro can do this, you must understand how the Before Update property works.

About the Before Update property

Using the Before Update property is the best way to validate control entries. Why? Well, the Before Update event occurs as you try to leave the control but before actually updating the field entry. If the macro you assign to this property issues the CancelEvent action, it will disable your attempt to leave the control. Therefore, the macro can perform a validation test and keep you from leaving the field if the entry fails the test.

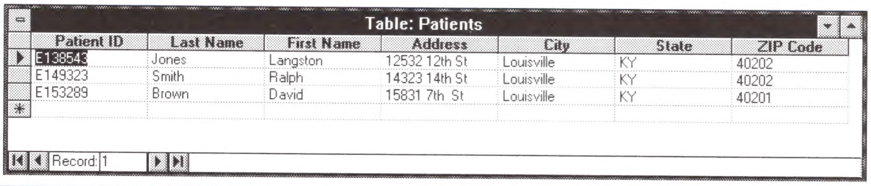
In our example, we'll assign a macro to the Before Update property of the primary field's control. The macro will use DCount( ) to check for a duplicate key value. If it finds one, it'll prevent you from leaving the key field's control until you supply a valid key value. If you enter a unique value, the macro will let you move to the next field normally.

An example

Let's examine this technique in terms of a specific example. Suppose you enter the names and addresses of new patients into the Patients table shown in Figure B. When you add new patients, you assign them a unique Patient ID number. The Patients table uses this number as its key field.

Next, let's create the form you'll use for adding and editing patient data. Highlight the Patients table in the Database window and click the New Form (▤) button on

Figure B



Patient ID	Last Name	First Name	Address	City	State	ZIP Code
E138543	Jones	Langston	12532 12th St	Louisville	KY	40202
E149323	Smith	Ralph	14323 14th St	Louisville	KY	40202
E153289	Brown	David	15831 7th St	Louisville	KY	40201

You want to create a macro that prevents key violations while adding and editing data in the Patients table.

Inside MICROSOFT ACCESS™

Inside Microsoft Access (ISSN 1067-8204) is published monthly by The Cobb Group.

Prices:

Domestic	$59/yr. ($7.00 each)
Outside US	$79/yr. ($8.50 each)

Phone:

Toll free	(800) 223-8720
Local	(502) 491-1900
Customer Relations Fax	(502) 491-8050
Editorial Department Fax	(502) 491-4200

Address:
You may address tips, special requests, and other correspondence to

The Editor, *Inside Microsoft Access*
9420 Bunsen Parkway, Suite 300
Louisville, KY 40220

For subscriptions, fulfillment questions, and requests for bulk orders, address your letters to

Customer Relations
9420 Bunsen Parkway, Suite 300
Louisville, KY 40220

Advertising: For information about advertising in Cobb Group journals, contact Tracee Bell Troutt at (800) 223-8720, ext. 430.

Postmaster:
Second class postage is pending in Louisville, KY. Send address changes to

Inside Microsoft Access
P.O. Box 35160
Louisville, KY 40232

Authorized Canada Post International Publications Mail (Canadian Distribution) Sales Agreement #XXXXXX CANADA GST #123669673. Send returns to Canadian Direct Mailing Sys. Ltd., 920 Mercer Street, Windsor, Ontario, N9A 7C2. Printed in the USA.

Staff:

Editor-in-Chief	David Brown
Editing	Timothy E. Hampton Elizabeth Welch
Production Artist	Julie Jefferson
Design	Karl Feige
Publications Manager	Tara Dickerson
Managing Editor	Suzanne Thornberry
Circulation Manager	Brent Shean
Publications Director	Linda Baughman
Editorial Director	Jeff Yocom
Publishers	Mark Crane Jon Pyles

Back Issues:
To order back issues, call Customer Relations at (800) 223-8720. Back issues cost $7 each, $8.50 outside the US. You can pay with MasterCard, VISA, Discover, or American Express, or we can bill you. Please identify the issue you want by the month and year it was published. Customer Relations can also provide you with an issue-by-issue listing of all articles that have appeared in *Inside Microsoft Access*.

the tool bar. Click the FormWizards button in the New Form dialog box. Then, in the next dialog box, select *Tabular* and click OK. In the dialog box that follows, just click the Fast Forward button (⏭). In the wizard's last dialog box, click the Design button. The wizard will generate the form shown in Figure C.

You're now ready to create the macro that checks for duplicate key values. But first, check the name of the Patient ID field's control, since you'll need to know the control's name as you create the macro. Click the Patient ID field's text box and then click the Properties button (▦) on the tool bar. You'll find that the wizard automatically assigned the control name *Patient ID* to the Control Name property, as shown in Figure D.

Creating the macro

To create the macro, return to the Database window by pressing [F11]. Next, display the list of macros by clicking the Macro button and then click the New button. When the new Macro window appears, open the Condition column by clicking the Conditions button (▦) on the tool bar. Then, enter the conditions and actions listed in Table A. Remember to include the elipses in the second row's conditions column.

When you've finished, pull down the File menu and select the Save As... command. In the Save As dialog box, enter *Patient Key Violation Test* and click OK. Then, close the Macro window.

Assigning the macro to the Patient ID text box

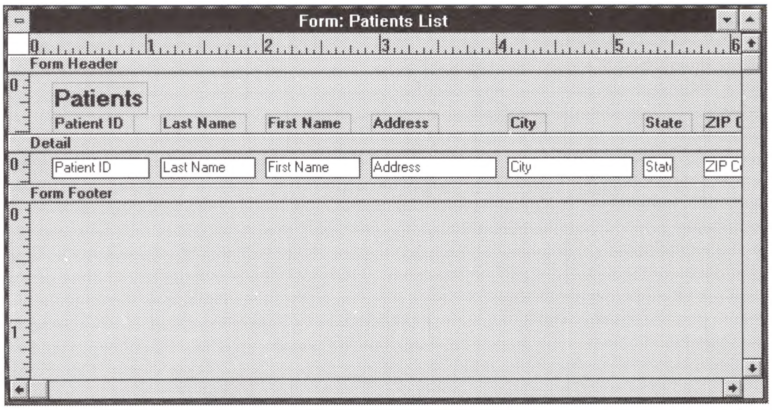
Finally, you're ready to assign the macro to the Patient ID field's text box control. Move to the form's window and make sure the property sheet lists the Patient ID text box's properties. Then, move to the Before Update property, click its dropdown arrow, and select the Patient Key Violation Test macro from the list, as shown in Figure E.

Now test the macro's operation by clicking the Form View button (▦) on the tool bar. Move to the Patient ID control and duplicate an existing Patient ID entry. The error message we showed you in Figure A on page 1 will appear. When you click OK, the cursor will still be in the Patient ID text box. You won't be able to leave until you supply a unique entry. You can undo your changes to the entry by pressing the [Esc] key.

Conclusion

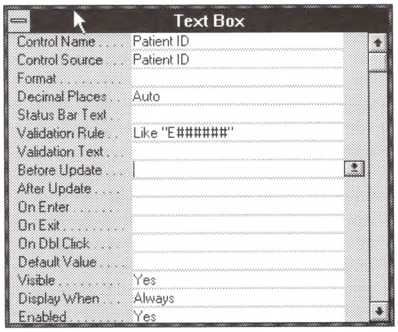
In this article, we showed you how to create a macro that checks for duplicate entries in your table's primary key field during data entry. You can assign the macro to the key field control's Before Update property so that the macro checks for duplicate entries as soon as you try to leave the field. ◆

Figure C



You create this default tabular form for testing our technique.

Figure D

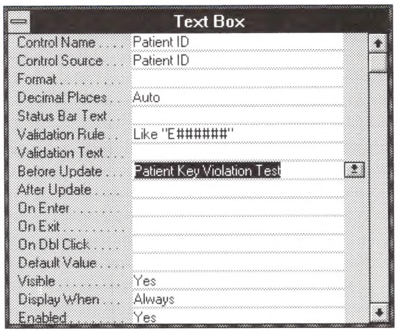


Text Box

Control Name	Patient ID
Control Source	Patient ID
Format	
Decimal Places	Auto
Status Bar Text	
Validation Rule	Like "E######"
Validation Text	
Before Update	
After Update	
On Enter	
On Exit	
On Dbl Click	
Default Value	
Visible	Yes
Display When	Always
Enabled	Yes

Notice that Access by default assigns the Control Name property to the name of the field.

Figure E



Text Box

Control Name	Patient ID
Control Source	Patient ID
Format	
Decimal Places	Auto
Status Bar Text	
Validation Rule	Like "E######"
Validation Text	
Before Update	Patient Key Violation Test
After Update	
On Enter	
On Exit	
On Dbl Click	
Default Value	
Visible	Yes
Display When	Always
Enabled	Yes

To trap key violations, assign the Patient Key Violation Test macro to the Patient ID text box's Before Update property.

Table A

Conditions	Actions	Action Properties

DCount("[Patient ID]","Patients", "[Patient ID] = Form![Patient ID]")	MsgBox	Message = Enter a unique Patient ID
		Beep = Yes
		Type = Critical
		Title = Error!

...	CancelEvent	

Access Basic

Preventing users from adding new records while editing data

If you're like most users, you probably create different forms for the many tasks involved in managing your database. For instance, you might create separate forms for adding new records to a table and editing data in existing records.

To let you control the editing capabilities of your forms, Access provides a form property called Default Editing. You assign to this property the value *Allow Edits* for forms that allow editing; you set the property to *Data Entry* for data-entry forms. The *Read Only* setting is reserved for forms that restrict editing entirely.

At first, you might think these three settings provide all the data-entry modes you'd ever need. However, one important editing mode is missing. None of these settings lets you only *edit* existing data. When you assign the *Allow Edits* value to the Default Editing property, you can *add* records to the form as well as edit existing records.

In many cases, you'll want to be able to add records while editing. If so, the default operation will suit your purposes. However, if you want to separate data entry from data editing, you must take steps to remove from the normal editing mode the ability to add records. In this article, we'll show you how to create a simple Access Basic function called PreventNewRecords( ) that will prevent you from adding records when you're using the Allow Edits mode.

A brief overview

Let's start with a brief discussion of how the function works and how you can incorporate it into your forms. In a nutshell, PreventNewRecords( ) disables the blank record at the end of the table or query dynaset. The function detects when you try to move to the blank record and then immediately returns the cursor to the last record in the table or dynaset. The net result is that you'll never be able to move past the last existing record in the table.

You assign the function to the form's On Current property. As you may know, the On Current event occurs every time you arrive in a new record. Therefore, every time you move to a new record, the function will check whether you've moved to the blank record and kick you out when you do.

Background information

If you're unfamiliar with Access Basic programming, you may not immediately understand how the function works. But don't worry: You'll easily be able to follow along after we describe just two Access Basic features. To comprehend this function, you must understand how the On Error statement lets your function respond to runtime errors. You must also know about the Bookmark property. We'll briefly discuss these points; then, we'll show you the function.

Error handling in Access Basic

Let's start with Access Basic's error-handling features. You use the On Error command to tell the function how to respond to runtime errors, and you use the Err function to determine when an error actually occurs.

On Error has several options, but we'll use the Resume Next option. The *On Error Resume Next* statement tells the function to simply continue if an error occurs. When you use this option, you must check for problems after every line you think might cause an error. That way, when an error occurs, the function will respond to the error before it does anything else.

You check for errors by using the Err function. The Err function returns a number that identifies the error that occurs. If you call Err when an error has *not* occurred, the function returns 0. Before we show you how you use these error-handling features, we'll discuss the Bookmark property.

The Bookmark property

The other Access Basic feature you use in the PreventNewRecords( ) function is the Bookmark property. You probably don't realize it, but every record has a Bookmark property value that Access can use to return to the record quickly. In an Access Basic function or procedure, you can store a record's bookmark in a string variable. Later in the program, you can return to that record by using the bookmark.

ORDER TODAY!

Is This Your Copy?

If Not, Order Your Subscription Today!

☑ Yes! Enter my subscription for one year of *Inside Microsoft Access* for only $49. That's just $4.08 a copy instead of the usual $6 an issue!
Please bill me later.

Name ____

Firm ____

Address ____

City ______________________ State _________ Zip _________

Phone (_________)_________________________ S 2066

Outside U.S.: $8.50 ea., $69/yr. VCJ1

Inside
MICROSOFT ACCESS

NEW ORDER!

NO POSTAGE
NECESSARY
IF MAILED IN
THE UNITED
STATES

BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 618 LOUISVILLE, KENTUCKY

POSTAGE WILL BE PAID BY ADDRESSEE

THE COBB GROUP
PO BOX 35160
LOUISVILLE KY 40232–9719

For our purposes, we don't care about using bookmarks for moving among the records. Instead, we're interested in an obscure aspect of bookmarks. The blank row at the bottom of the datasheet window does *not* have a bookmark. This is important because, if you try to access the Bookmark property for that row, an error will occur.

At this point, you might be able to guess how PreventNewRecords( ) will work. You tell the function to watch for errors and then issue a statement that will generate an error whenever you move into the blank record. The Bookmark property just provides a tricky way to generate the appropriate error.

The PreventNewRecords( ) function

Now let's create the new function. In the Database window, click the Module button and then the New button. When the new Module window appears, pull down the Edit menu and select the New Procedure... command. Next, in the New Procedure dialog box, make sure you've selected the Function radio button in the Type section; then, enter *PreventNewRecords* in the Name text box and click OK.

The Module window will then display an empty function definition. You type the function's statements between the Function and the End Function statements, as shown below:

```
Function PreventNewRecords ( )
 Dim a
 On Error Resume Next
 a = Screen.ActiveForm.Bookmark
 If Err <> 0 Then DoCmd GoToRecord , , A_LAST
End Function
```

Next, pull down the Run menu and select the Compile All command. Then, save the new module by using the Save command from the File menu. Enter *Prevent New Records* in the dialog box and click OK. Finally, close the Module window.

Now let's see how PreventNewRecords( ) works. First, the function declares the variable *a*. It then executes the *On Error Resume Next* statement that tells the function to continue running when errors occur.

The next statement tries to assign the current record's Bookmark property to the variable *a*. This statement will execute well for all cases except one—when the current record is the blank record at the end of the table or dynaset. The last statement in the function tests the Err function to see whether the previous statement caused an error. If so, the DoCmd GoToRecord command moves the cursor to the last existing record.

Using the PreventNewRecords( ) function in a form

Let's create a simple form for the NWIND database's Products table, which stores the items that the Northwind Company sells. Before continuing, use the Import… command on the File menu to create a copy of the Products table in your database. Figure A shows a screenful of data; the entire table contains over 70 records.

Figure A

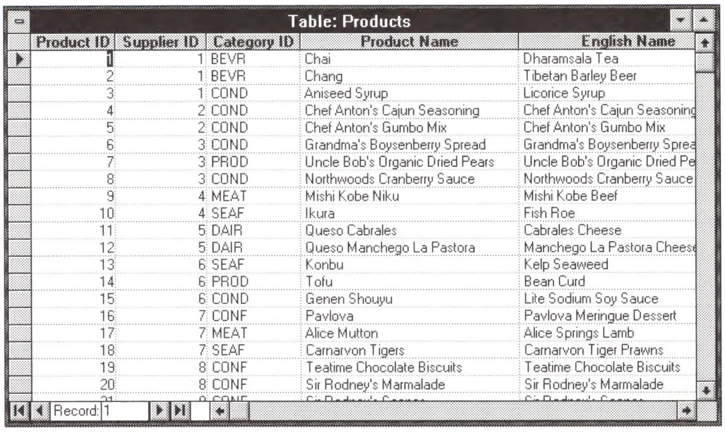


Table: Products

Product ID	Supplier ID	Category ID	Product Name	English Name
1	1	BEVR	Chai	Dharamsala Tea
2	1	BEVR	Chang	Tibetan Barley Beer
3	1	COND	Aniseed Syrup	Licorice Syrup
4	2	COND	Chef Anton's Cajun Seasoning	Chef Anton's Cajun Seasonin
5	2	COND	Chef Anton's Gumbo Mix	Chef Anton's Gumbo Mix
6	3	COND	Grandma's Boysenberry Spread	Grandma's Boysenberry Sprea
7	3	PROD	Uncle Bob's Organic Dried Pears	Uncle Bob's Organic Dried Pe
8	3	COND	Northwoods Cranberry Sauce	Northwoods Cranberry Sauce
9	4	MEAT	Mishi Kobe Niku	Mishi Kobe Beef
10	4	SEAF	Ikura	Fish Roe
11	5	DAIR	Queso Cabrales	Cabrales Cheese
12	5	DAIR	Queso Manchego La Pastora	Manchego La Pastora Chees
13	6	SEAF	Konbu	Kelp Seaweed
14	6	PROD	Tofu	Bean Curd
15	6	COND	Genen Shouyu	Lite Sodium Soy Sauce
16	7	CONF	Pavlova	Pavlova Meringue Dessert
17	7	MEAT	Alice Mutton	Alice Springs Lamb
18	7	SEAF	Carnarvon Tigers	Carnarvon Tiger Prawns
19	8	CONF	Teatime Chocolate Biscuits	Teatime Chocolate Biscuits
20	8	CONF	Sir Rodney's Marmalade	Sir Rodney's Marmalade

We'll create a form for the Products table that prevents you from adding records while letting you edit existing records.

Since you add data to this table only when the Northwind Company decides to sell additional products, you want to withhold the table's data. In other words, you want to create a form that allows you to edit existing product data but refuses to let you add any more products.

Start by creating a new form for the Products table. Highlight the table in the Database window and click the New Form button (▤) on the tool bar. In the New Form dialog box, click the FormWizards button. In the dialog box that follows, highlight the *Single-Column* item and click OK. When the form wizard's first dialog box appears next, create a default form by clicking the Fast Forward button (⏭). In the final dialog box, click the Design button. Access will generate the form shown in Figure B on the next page.

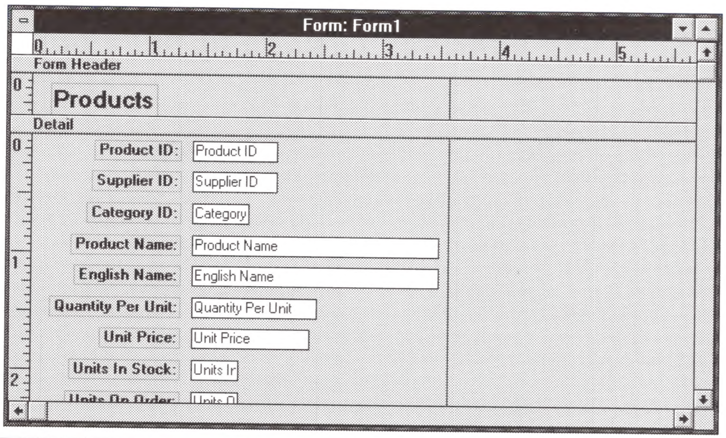
Now set up the call to the function PreventNewRecords( ). Click the Property Sheet button (▤) on the tool bar while the form is still selected. If you've inadvertently selected another control or section, click the white square at the intersection of the rulers to select the form again.

Once the property sheet is onscreen and listing the form properties, move to the On Current property and enter

```
=PreventNewRecords( )
```

Be sure to begin with the equal sign (=). If you don't, Access will think PreventNewRecords( ) is a macro, and an error will occur.

Figure B



Form: Form1

Form Header

Products

Detail

Product ID: Product ID

Supplier ID: Supplier ID

Category ID: Category

Product Name: Product Name

English Name: English Name

Quantity Per Unit: Quantity Per Unit

Unit Price: Unit Price

Units In Stock: Units Ir

Units On Order: Units O

This form will use the PreventNewRecords( ) function to prevent data entry.

You're now ready to use the form. You can best see the effect of the function when you use the form in Datasheet view. When you click the Datasheet button (▦) on the tool bar, the Products table will appear. Next, move to the bottom of the datasheet and try to move into the blank row by pressing the ⬇ key. The cursor will stay put.

Using the function with many forms

The nice thing about this function is that it's completely generic—you can use it with any form to prevent users from adding data. You can define it in all your databases and use it in any of your data-editing forms. As long as you assign the function to the form's On Current property and also set the Default Editing property to Allow Edits, the form will prevent you from adding data.

Conclusion

In this article, we showed you a way to modify a form's Allow Edits editing mode so that the form prevents you from adding data to a table. You create the PreventNewRecords( ) function we showed you and then simply assign the function to your form's On Current property. ◆

Access Tip

Context-sensitive Help is just a mouse click away

If you're like many users, you usually open your Access manuals as a last resort. Access' Help feature provides just about all the information you can find in the manuals, and the Help feature packages the information in a friendly and flexible way. The Help system lets you easily search for the Help topic you need and then shift among various related Help topics.

You can find almost all the information you need by using the Help system's search feature. However, the process isn't always straightforward. You often need to browse a couple of Help topics before you find the particular Help screen you need.

Fortunately, Access provides a context-sensitive Help that can immediately deliver the Help screen for the object or control you're working with. In this article, we'll describe the two methods for using context-sensitive Help.

Obtaining context-sensitive Help

Context-sensitive Help became standard for most applications after Windows 3.1 shipped. You simply select the control or object for which you want help and then press [F1]. A Help window will appear showing the information on that control or object.

This method works fine in almost every situation. However, Access provides another way to obtain context-sensitive Help. When you press [Shift][F1], the mouse pointer will change from the ordinary pointer to a pointer with a question mark behind it. You can then place the new pointer over the Access object or control and click. The standard Access Help information on that object or control will appear as it does when you use the other method.

An example

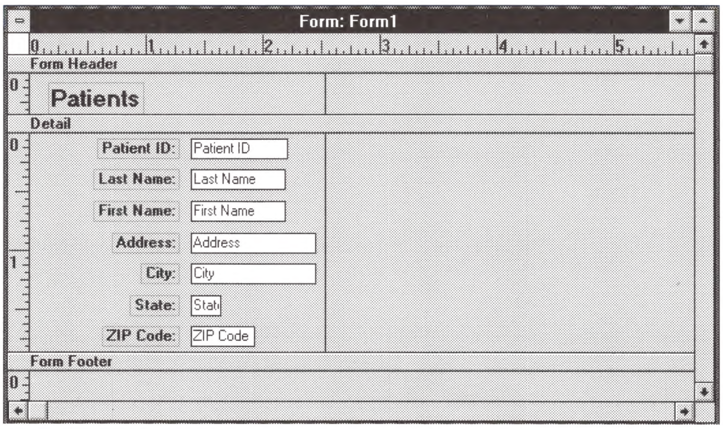
Let's look at a simple example. Suppose you're designing the form shown in Figure A and you want to create a validation rule for one of the text box controls. If you aren't sure how to set validation rules, you'll want to see the Help application's information concerning validation rules.

You know you create validation rules by setting the Validation Rule property for the control, so you first select the text box control and then open the property sheet. The Validation Rule property will appear in the list.

Now, if you want more information about validation rules, you have two choices. By using the conventional method, you can move to the Validation Rule property in the property sheet and then press the [F1] key. Alternatively, you can press [Shift][F1] to obtain the Help pointer and then click in the Validation Rule text box in the property sheet.

Figure B shows the Help pointer that appears when you use the second method. After you click in the Validation Rule text box with the mouse pointer, the Help screen will appear, as shown in Figure C. You can then use the Help text to formulate the validation rule for the control. ◆

Figure A

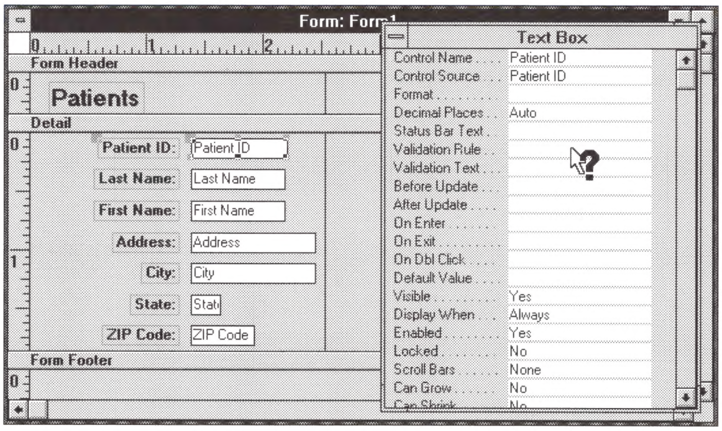


Patients

Patient ID:	Patient ID
Last Name:	Last Name
First Name:	First Name
Address:	Address
City:	City
State:	Stat
ZIP Code:	ZIP Code

You want to define a validation rule for a text box on this form.

Figure B



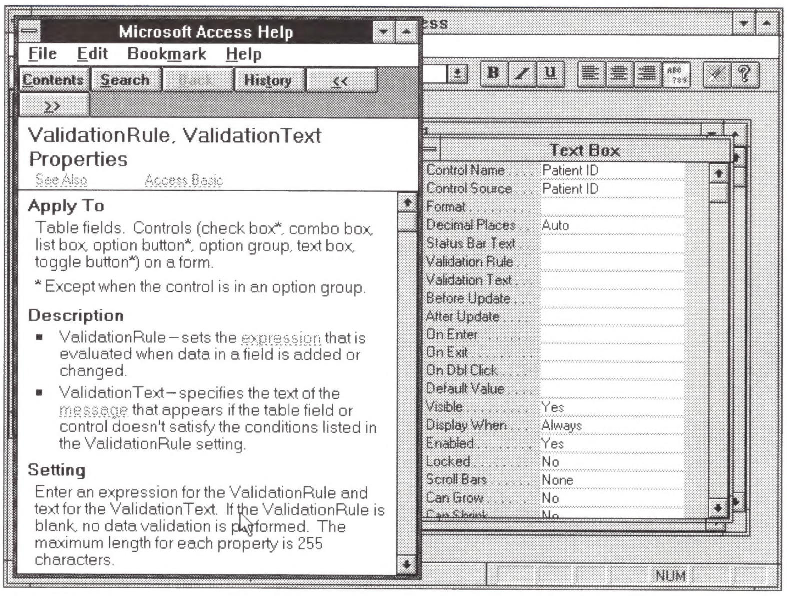
Patients

Patient ID:	Patient ID
Last Name:	Last Name
First Name:	First Name
Address:	Address
City:	City
State:	Stat
ZIP Code:	ZIP Code

Control Name	Patient ID
Control Source	Patient ID
Format	
Decimal Places	Auto
Status Bar Text	
Validation Rule	
Validation Text	
Before Update	
After Update	
On Enter	
On Exit	
On Dbl Click	
Default Value	
Visible	Yes
Display When	Always
Enabled	Yes
Locked	No
Scroll Bars	None
Can Grow	No
Can Shrink	No

Pressing [Shift][F1] provides the Help pointer.

Figure C



ValidationRule, ValidationText Properties

See Also   Access Basic

Apply To

Table fields. Controls (check box*, combo box, list box, option button*, option group, text box, toggle button*) on a form.

* Except when the control is in an option group.

Description

- ValidationRule – sets the expression that is evaluated when data in a field is added or changed.
- ValidationText – specifies the text of the message that appears if the table field or control doesn't satisfy the conditions listed in the ValidationRule setting.

Setting

Enter an expression for the ValidationRule and text for the ValidationText. If the ValidationRule is blank, no data validation is performed. The maximum length for each property is 255 characters.

Control Name	Patient ID
Control Source	Patient ID
Format	
Decimal Places	Auto
Status Bar Text	
Validation Rule	
Validation Text	
Before Update	
After Update	
On Enter	
On Exit	
On Dbl Click	
Default Value	
Visible	Yes
Display When	Always
Enabled	Yes
Locked	No
Scroll Bars	None
Can Grow	No
Can Shrink	No

Clicking the Validation Rule property with the Help pointer invokes the validation-rule Help screen.

Editing

Copying the previous record's entry saves time during data entry

Have you ever been entering addresses into a table and had to enter the same city and state in every record? There's nothing worse than entering a city name like *Louisville* for ten addresses in a row.

Fortunately, Access has a little-known keystroke shortcut that eliminates this problem. When you press [Ctrl]' (holding down the [Ctrl] key and pressing the apostrophe key), Access duplicates in the current field the previous record's entry in that field. By using this feature, you don't need to type that long city name in every record. As long as you're entering a series of addresses for the same city, you can use the keystroke shortcut when entering the City field—and the State field, for that matter.

Figure A helps us illustrate how you might be able to use this technique. The figure shows the Patients table, into which you need to enter the names and addresses of several new patients.

Suppose you need to enter the information we list in Table A. After you've entered the first patient's data, you can use the [Ctrl]' key to duplicate field entries (such as *Louisville* in the City field and *KY* in the State field) for the other patients.

Note that this keystroke shortcut isn't designed for any particular circumstance. The [Ctrl]' key can come in handy in a variety of situations. For instance, Figure B shows how the Patients table will look when you're entering the fourth record. The fourth patient happens to live on the same street as the third. While the full address field isn't identical, you can press [Ctrl]' to copy the field entry and then simply change the street number.

Figure A

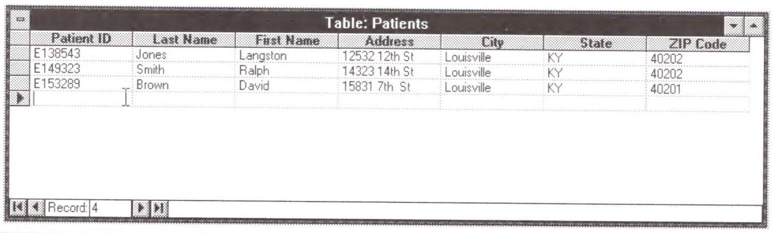


Table: Patients

Patient ID	Last Name	First Name	Address	City	State	ZIP Code
E138543	Jones	Langston	12532 12th St	Louisville	KY	40202
E149323	Smith	Ralph	14323 14th St	Louisville	KY	40202
E153289	Brown	David	15831 7th St	Louisville	KY	40201

Record: 4

Suppose your Patients table contains these patients' data.

Figure B

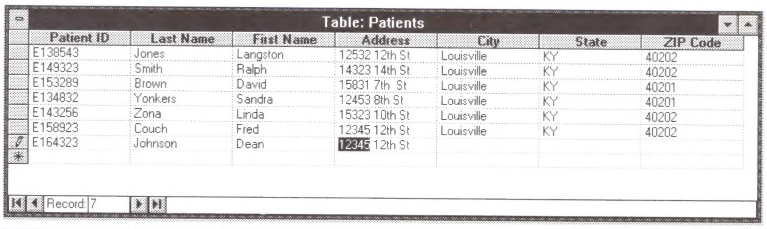


Table: Patients

Patient ID	Last Name	First Name	Address	City	State	ZIP Code
E138543	Jones	Langston	12532 12th St	Louisville	KY	40202
E149323	Smith	Ralph	14323 14th St	Louisville	KY	40202
E153289	Brown	David	15831 7th St	Louisville	KY	40201
E134832	Yonkers	Sandra	12453 8th St	Louisville	KY	40201
E143256	Zona	Linda	15323 10th St	Louisville	KY	40202
E158923	Couch	Fred	12345 12th St	Louisville	KY	40202
E164323	Johnson	Dean	12345 12th St			

Record: 7

You can copy field entries with the [Ctrl]' keystroke shortcut in order to modify the previous record's entry.

Table A

Data for the Patients Table

Patient ID	Name	Address	City, State, and ZIP
E134832	Sandra Yonkers	13453 8th St	Louisville, KY 40201
E143256	Linda Zona	15323 10th St	Louisville, KY 40202
E158923	Fred Couch	12345 12th St	Louisville, KY 40202
E164323	Dean Johnson	15343 12th St	Louisville, KY 40202

Notes

If you've worked with Borland's database managers, either Paradox for DOS or Paradox for Windows, you're accustomed to having this data-entry feature. Paradox's Ditto key copies the previous record's entry as well. However, Access' implementation of this keystroke shortcut has one important difference: It will always copy the previous record's entry, regardless of whether you've begun entering text in the current record's field. On the other hand, Paradox's Ditto key won't copy the previous record's entry unless the current record's field is empty. ◆

Microsoft Access™ Spotlight

Targeted Advertising

The Cobb Group
December 1993

Unwrap the secrets to Access...

Total ACCESS™

Database Documentation Program

for Microsoft Access®

Supports Access 1.0 & 1.1

Comprehensive Documentation and Analysis of Microsoft Access® Databases!

Microsoft Access offers a sophisticated new way to develop Windows applications. You can easily build complex forms, reports, queries and macros. However, with this power comes complexity. As you add features to your applications, understanding them becomes more difficult. Total Access provides the documentation you need to take control of your databases. From simple features such as printing macros and table structures to more complex documentation such as macro calls and fields used across objects, Total Access allows you to create Microsoft Access applications more efficiently.

Documents:

- Tables
- Fields
- Queries
- Forms
- Reports
- Controls
- Field Cross-Reference
- Table Relationships
- Macros and Calls
- Object Inventory
- Descriptions
- Properties

and more!

Features:

- Entirely menu driven.
- Runs from within Microsoft Access
- Documents all objects in one pass.
- All documentation placed in Access tables.
- View data using our sophisticated forms.
- Includes over 40 built-in reports.

Total Access provides the information you need to develop and maintain Access applications more productively.

Total Access places all output in Access tables so you can use Access to manage the documentation. View, sort or query the results. Use our reports or create your own!

> "Total Access is an indispensable utility that comprehensively documents Access databases....Every serious Access user should own a copy.... Total Access may just be the Access add-on of the year."
>
> Paul Litwin, Editor, *Smart Access*

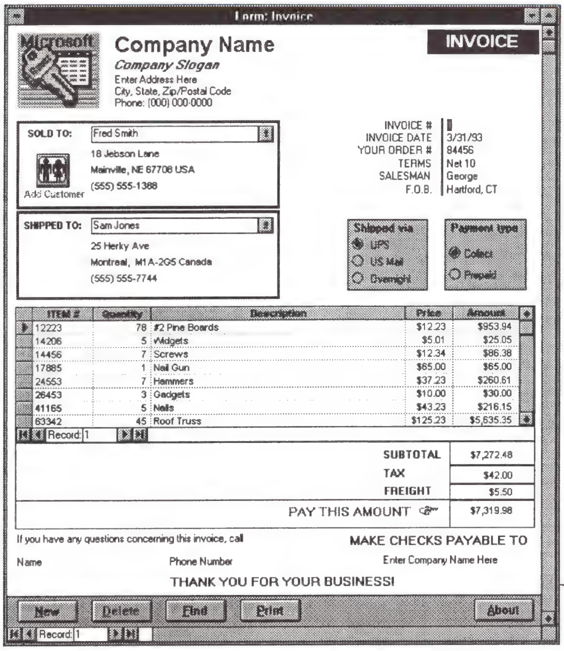
Total Access is available now at the introductory price of $145 + $5 S&H ($15 Int'l). That's $50 off the regular price of $195. Virginia residents add 4.5% sales tax.

30 day money back guarantee!

FMS

FMS, Inc.
8027 Leesburg Pike
Suite 410, Dept. 322
Vienna, VA 22182, USA
(703) 356-4700 ext. 322
FAX (703) 448-3861, Compuserve 73710,463

THE BEST ACCESS ADD-ONS!



Company Name
Company Slogan
Enter Address Here
City, State, Zip/Postal Code
Phone: (000) 000-0000

INVOICE

SOLD TO: Fred Smith
18 Jebson Lane
Mainville, NE 67708 USA
(555) 555-1388

INVOICE #	
INVOICE DATE	3/31/93
YOUR ORDER #	84456
TERMS	Net 10
SALESMAN	George
F.O.B.	Hartford, CT

SHIPPED TO: Sam Jones
25 Herky Ave
Montreal, M1A-2G5 Canada
(555) 555-7744

Shipped via: ● UPS ○ US Mail ○ Overnight

Payment type: ● Collect ○ Prepaid

ITEM #	Quantity	Description	Price	Amount
12223	78	#2 Pine Boards	$12.23	$953.94
14206	5	Widgets	$5.01	$25.05
14456	7	Screws	$12.34	$86.38
17885	1	Nail Gun	$65.00	$65.00
24553	7	Hammers	$37.23	$260.61
26453	3	Gadgets	$10.00	$30.00
41165	5	Nails	$43.23	$216.15
63342	45	Roof Truss	$125.23	$5,635.35

Record: 1

SUBTOTAL	$7,272.48
TAX	$42.00
FREIGHT	$5.50
PAY THIS AMOUNT	$7,319.98

If you have any questions concerning this invoice, call
Name Phone Number

MAKE CHECKS PAYABLE TO
Enter Company Name Here

THANK YOU FOR YOUR BUSINESS!

Record: 1

The Access Business Forms Library

This is a library of professionally designed business forms that you can use in a standalone Access database container or integrate into your own Access applications. You can distribute individual forms from the library royalty free.

Each application includes all tables, forms, reports, and macros for instant use! Because each table, form, and report are standard Access objects, you can quickly and easily customize them to your own needs. Also included is a 120 page detailed user guide describing business uses of each form as well as a picture of each form and a customization guide to teach you how each form was created.

Special Price *Only* $79.95

No Royalty Use Fees!

Sample List Includes:
About Business Forms
Card and Gift Tracker
Cash Receipts
Check Requisition Form
Commission Report
Conference Room Scheduler
Contributions Tracker
Credit Application
Customer Registration
Expense Form
Final Notice
Back Order Letter
Past Due Notice
PC Tracker
Address List
Personnel Record
Price List
Prospecting Form
Purchase Order
Quotation
Receiving Report
Credit Reminder
Resume Receipt
Household Inventory
Invoice
Job Listing
Job Application
Bill of Lading
Telephone Messages
RFP/RFQ
Sales Order
Seminar Registration
Shipping Order
Videotape Log
Wedding Planner

The Mail Merge Report Wizard

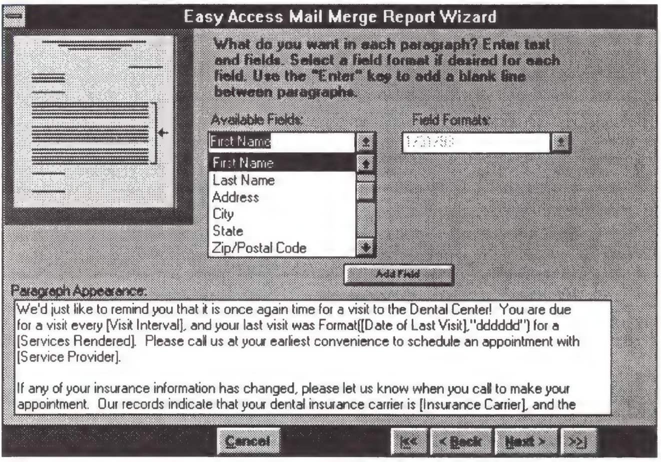
The Wizard that Microsoft Forgot!

- Create Form Letters in Access in Eight Easy Wizard Screens Without A Word Processor

"Makes Creating Mail-Merge Letters Easy for Anyone Without Having to Use a Word Processing Package"
Michael Irwin - Access Book Author

- Works Like Other Access Report Wizards
- Customize the Completed Mail-Merge Letter
- Create Multi-Page Letters
- Creates a Standard Access Report
- Complete Documentation

Only $39.95



What do you want in each paragraph? Enter text and fields. Select a field format if desired for each field. Use the "Enter" key to add a blank line between paragraphs.

Available Fields: First Name
First Name
Last Name
Address
City
State
Zip/Postal Code

Field Formats:

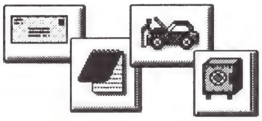
Paragraph Appearance:
We'd just like to remind you that it is once again time for a visit to the Dental Center! You are due for a visit every [Visit Interval], and your last visit was Format([Date of Last Visit],"dddddd") for a [Services Rendered]. Please call us at your earliest convenience to schedule an appointment with [Service Provider].

If any of your insurance information has changed, please let us know when you call to make your appointment. Our records indicate that your dental insurance carrier is [Insurance Carrier], and the

Also available...

The Button Bundle

1200 Command Buttons for Customizing Access Forms!



NEW! Special Price, $99.95

See our other ad in Inside Access for special offers on our books and training videos!

US Phone Orders: (800) 277-3117

□ **Yes! I want to order.**

□ No. I don't wish to order at this time. Please send me detailed information on these products at no charge.

Qty	Item	Price	Total
___	Access Business Forms Library	$79.95	______
___	The Mail Merge Report Wizard	$39.95	______
___	The Button Bundle	$99.95	______
___	Forms and Buttons Sampler	$ 9.95	______
		Subtotal:	______
		Shipping & Handling:	$4.95*
		Total:	______

MICROSOFT CONSULTING CHANNEL
Partner

* Canadian Orders: Add $10.00
* Europe/S. America: Add $25.00
* Pacific Rim/Africa: Add $35.00

Fax and International Orders Call (203) 644-5891

Mail/Fax Orders: Fill in the Form Below and Send To:
Cary Prague *Books and Software*
60 Krawski Dr.
S. Windsor, CT 06074

Name____
Company_____________________________________
Address______________________________________
City_____________________State____Zip__________
Country___________________Phone______________
Payment: □Check □Visa □MasterCard □Amex
Card#________________________Exp. Date________

Signature_____________________________________

All Software Shipped 2nd Day
Overnight Shipping Available

WIND

Windows Information Network for Developers

Join the only BBS dedicated to supporting developers of Access, Visual Basic, C++ and SQL.

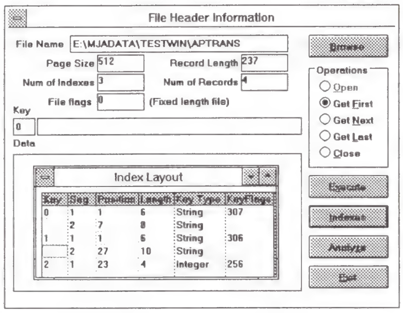
WIND offers: a full messaging system for questions, tips and tricks; file areas filled with freeware, shareware and demo programs; and, developers' books and commercial software products.

To take a free look at WIND, dial-in to **(216) 694-5734**.

WIND
200 Public Square, Ste. 26-4600
Cleveland, Ohio 44114

Visual DDF for Btrieve

Do you have a Btrieve data file and don't know the layout? Do you want to create the data dictionary (DDF) files and attach to Access or VB 3.0? Visual DDF from Prodata can help.



File Header Information

File Name: E:\MJADATA\TESTWIN\APTRANS

Page Size: 512 Record Length: 237

Num of Indexes: 3 Num of Records: 4

File flags: 0 (Fixed length file)

Key: 0

Data

Operations: ○ Open ● Get First ○ Get Next ○ Get Last ○ Close

Index Layout

Key	Seg	Position	Length	Key Type	KeyFlags
0	1	1	6	String	307
	2	7	8	String	
1	1	1	6	String	306
	2	27	10	String	
2	1	23	4	Integer	256

- **File Specifications**
  - Page Size
  - Record Length
  - Number of Indexes
  - Number of Records
  - File Flags
  - Fixed & Variable
- **Index Specifications**
  - Multi-segment
  - Position & Length
  - Data Type
  - Key Flags

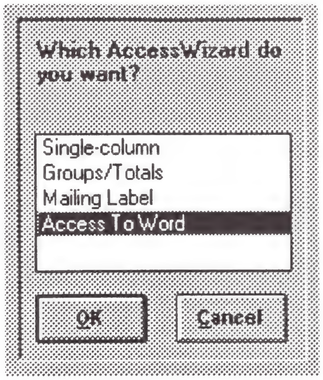
Any valid Btrieve data file can be opened. The file specifications and a complete layout of the indexes are displayed. Individual records in the file can be retrieved and analyzed to determine the type of data on a byte by byte basis. After the record layout has been mapped, you can create the standard *File*, *Field*, and *Index* DDF files. With these files, you can attach to Microsoft Access. *Special Spotlight price–$89.00.*

Prodata
12101 Menaul Blvd NE • Albuquerque, NM 87112
(505) 294-1530 • Fax (505) 298-3757

Access To Word - The #1 Access Add-on!

"Sheer Wizardry! Easy to use." Mike Mullin, MTX International, Inc.

Access To Word **merges data from your Microsoft® Access™ tables and queries to your Microsoft Word for Windows™ documents - quickly and easily!**



Which AccessWizard do you want?

Single-column
Groups/Totals
Mailing Label
Access To Word

- *Installs as an Access Wizard! From your Access database, simply choose File ⇨ New Report ⇨ ReportWizards ⇨ Access To Word!*
- *Also installs into Word's Print Merge feature! From your Word document choose File ⇨ Print Merge ⇨ Attach Data File!*
- *From Print Merge, allows you to select any Access table or query as the source, then apply any record selection criteria and sort levels!*
- *NEW! Includes developer API features. Distribute Access To Word runtime with your applications!*

"Links the best features of Word to my Access data!"
Helen Feddema, author

"Worth every penny." Peter Jackson

Access 1.1 Compatible

Kwery Corporation
Quality Database Tools

Order *Access To Word* today for only $79! (+ $5 S&H)
Call (800)AT-KWERY 24 hours a day. Order Code 93897.

P.O. Box 6726 Bellevue, WA 98008-0726 (206)644-7830 phone (206)644-8409 fax. Microsoft is a registered trademark and Access and Word for Windows are trademarks of Microsoft Corporation.

Video Training Systems

Master your software packages in record time using Softech's new *Video Based* Training Systems

Missing out on the advantages of Video-Based Training? Our highly acclaimed PC and technical courses - developed exclusively in-house - are packed with a wealth of information, hints and tips and will provide expert tuition. All courses are fully integrated with computer-based training exercises and course guide. A modular format offers the choice of completing the entire course from start to finish or selecting individual modules for a quick refresher. Even the most experienced user will find something new! Now you can train an entire company for one fixed cost, with no hidden extras.

Courses for Microsoft® Windows™- based Packages

Softech's Video Based Training Courses are all you need to rapidly gain confidence in the use of your desktop software packages. Starting with basics and elementary usage, the course progresses to advanced features such as programming, macros and DDE in the final video. Each course includes four video cassettes, dual resource diskettes and course guide.

Each Set only $295.00 + Delivery.

End-User Courses are available in the following Microsoft packages:

Access, Excel, Word, Windows 3.1 & PowerPoint.

Developers Master Series

Softech's **Master Series** Courses are designed by professional developers for today's programmer. Each video is packed with a wealth of tips and techniques, as well as demonstrations that will help you to use advanced programming features in no time at all. This is backed up by "hands-on" exercises that you can complete on your own PC - with full solutions and suggestions provided on the accompanying disk. The 'Master Series' Courses include eight video cassettes, each accompanied by course guide and resource diskette.

Each Set only $599.00. Master Series Courses:

FoxPro 2.5 for Windows and Visual Basic

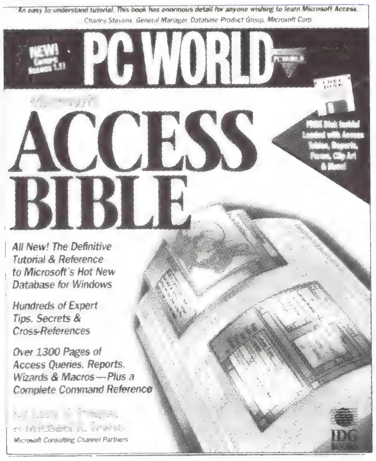
Softech Services Inc.

P. O. Box 4840, Winter Park, FL 32793 - Florida, USA.
Tel: (407) 678 8180 Fax: (407) 678 0086

Advertiser's Index

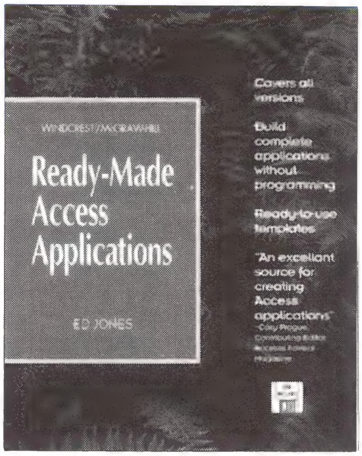
Microsoft Access SPOTLIGHT is published by The Cobb Group. For advertising rates and information, contact Tracee Bell Troutt (ext. 430) at (800) 223–8720.

The Best Access Books Today!


An easy to understand tutorial. This book has enormous detail for anyone wishing to learn Microsoft Access.
Charles Stevens General Manager, Database Product Group, Microsoft Corp
NEW!
PC WORLD
ACCESS BIBLE
All New! The Definitive Tutorial & Reference to Microsoft's Hot New Database for Windows
Hundreds of Expert Tips, Secrets & Cross-References
Over 1300 Pages of Access Queries, Reports, Wizards & Macros—Plus a Complete Command Reference
Microsoft Consulting Channel Partners
IDG BOOKS

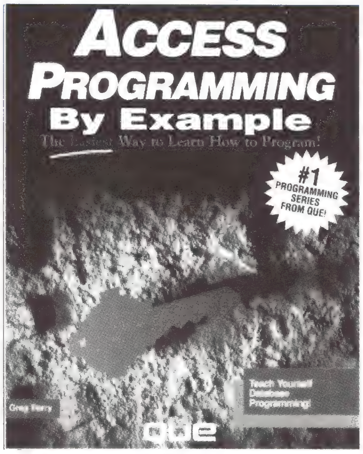
Access Bible
Prague/Irwin, IDG Books
1300 Pages, Includes Disk
Price: $34.95

Best Seller


WINDCREST/McGRAW-HILL
Ready-Made Access Applications
ED JONES
Covers all versions
Build complete applications without programming
Ready-to-use templates
"An excellent source for creating Access applications"

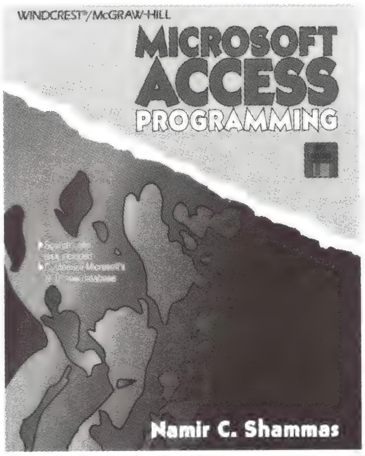
Ready Made Access Applications
Ed Jones, McGraw Hill
415 Pages, Includes Disk
Price: $36.95

NEW

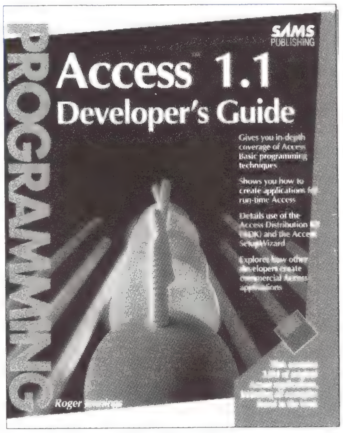

ACCESS PROGRAMMING By Example
The Easiest Way to Learn How to Program!
#1 PROGRAMMING SERIES FROM QUE!
Greg Perry
Teach Yourself Database Programming!
QUE

Access Programming by Example
Greg Perry, Que
600 Pages
Price: $27.95

NEW

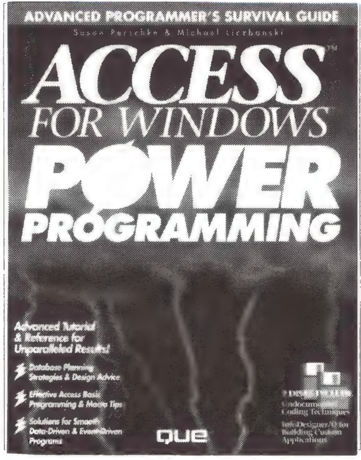


Microsoft Access Programming
Namir Shammas, McGraw Hill
300 Pages, Includes Disk
Price: $32.95



Access Developers Guide
Roger Jennings, Sams
1000 Pages, Includes Disk
Price: $37.95

NEW


ADVANCED PROGRAMMER'S SURVIVAL GUIDE
Susan Peschke & Michael Liczbanski
ACCESS FOR WINDOWS POWER PROGRAMMING
Advanced Tutorial & Reference for Unparalleled Results!
QUE

Access Power Programming
Liczbanski, Que
1000 Pages, Includes Disk
Price: $42.95

NEW

BONUS

Order any book on this page and get the bestselling 250 Access for Windows Hot Tips for only $12.50. This brand new 220 page book from Que, reveals tips and secrets about Access not found anywhere else.

Special!

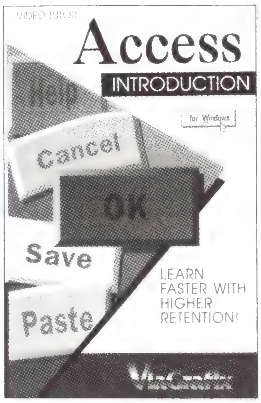
The Video Bundle

Introduction and Advanced

Both Videos...

Only

$89.95!



Each Video Also Includes a Disk!

US Phone Orders: (800) 277-3117

❒ **YES! I want to order.**

Qty	Item	Price	Total
___	Access Bible	$34.95	______
___	Ready Made Access Apps.	$36.95	______
___	Programming by Example	$27.95	______
___	Access Programming	$32.95	______
___	Access Developers Guide	$37.95	______
___	Access Power Programming	$42.95	______
___	The Access Video Bundle	$89.95	______
___	Bonus - 250 Access Hot Tips	$12.50	______
		Subtotal:	______
		Shipping & Handling:	$4.95 *
		Total:	______

MICROSOFT CONSULTING CHANNEL Partner

* Canadian Orders: Add $10.00 per book
* Europe/S. America: Add $25.00 per book
* Pacific Rim/Africa: Add $35.00 per book

Fax and International Orders Call (203) 644-5891

Mail/Fax Orders: Fill in the Form Below and Send To:

Cary Prague Books and Software
60 Krawski Dr.
S. Windsor, CT 06074

Name____________________________
Company_________________________
Address__________________________
City__________________State___Zip________
Country_______________Phone__________
Payment: ❒Check ❒ Visa ❒MasterCard❒ Amex
Card#__________________Exp. Date________
Signature__________________________

All books shipped US Mail Book Rate
Overnight Shipping Available

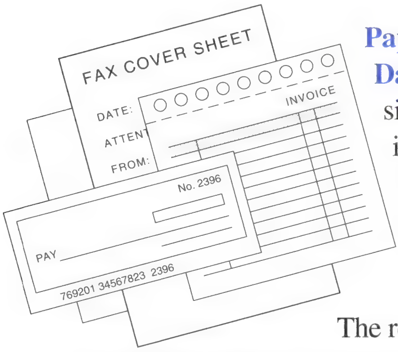
Wouldn't it be great to "copy" an image of a document right into a Microsoft Access database and then file it, annotate it, fax it, print it, and *share it* - using a software package that costs only ~~$245?~~

$145

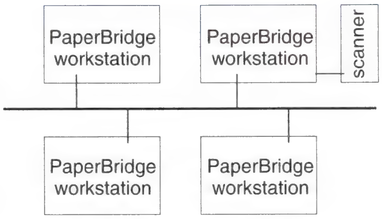
IMPOSSIBLE?

Document Image Management - On a PC! Now your desktop PC or workstation can perform document imaging tasks that previously required a much larger computer. Using new software technology you can copy documents into computer images and use them in place of paper. Routine office tasks can now be automated easily. Find documents faster and share them with other PC users. Make notes on them or store, fax, or e-mail them – all with PC document imaging management.

Microsoft ACCESS



PaperBridge for the Access Database – On the PC. It is simply the best value in document imaging available anywhere. PaperBridge adds full-function document image management capabilities to Microsoft's Access relational database. The result is industrial strength imaging seamlessly coupled with one of the most popular Windows database and application development environments. And PaperBridge offers other advanced features such as multi-page document display, user configurability, fast compression /decompression, annotation, and fax-in/fax-out. Altogether, a potent mix of features!



Powerful Peripheral and Network Support. PaperBridge is office-ready. It supports most network and Windows-based client/ server architectures. It also supports a full range of imaging peripherals and subsystems including scanners, faxes, printers, print accelerators, and optical storage systems.

PAPERBRIDGE
BUYERS ASSURANCE SEAL
INFO WORLD
September 1993

Easily Add Imaging to Access Applications. Between PaperBridge and Access, development of complete Windows-based imaging applications couldn't be easier – or less expensive. PaperBridge includes tools such as "wizards" and pre-tested macros to assist the novice developer as well as documentation and Access Basic routines for the experienced programmer. Access provides the rest. PaperBridge is the only open imaging development environment available. It's fast, complete and easy.

Ask for Your $145 Introduction to Our $245 Breakthrough. Until December 31, 1993 you can try the developer version of PaperBridge for only $145. The $100 savings is a great start to PC-based imaging with Access. One to a customer. Site and run-time license pricing available.

Check it out!

NOT IMPOSSIBLE WITH PAPERBRIDGE. TRY IT.

Make me a believer! Send me PaperBridge for Access – NOW.

To get your copy of PaperBridge for Access simply mail this coupon with your check for $145 plus $5.50 shipping and handling to:

TEAMWorks Technologies
65 Boston Post Rd. W., Marlboro, MA 01752

Or, for even faster service call toll-free:
1-800-532-3198

TEAMWorks
TECHNOLOGIES

NAME TITLE

COMPANY

STREET ADDRESS

CITY, STATE, ZIP,

PHONE NUMBER FAX NUMBER

VISA/MC □ AMEX □ DISCOVER □ Credit Card Number Expiration

SIGNATURE

Documenting your Import/Export specifications

Robert R. Smith contributed the material on which we based this article. Mr. Smith is an independent PC consultant and trainer in Microsoft Windows and Access. You may reach him at (612) 924-0439.

One of Access' strengths is the ease with which you can use external data. When you need to import or export data in a custom file format, you must first use the Imp/Exp Setup... command on the File menu to specify the data layout of the external file. Then, you can tell Access to use that file format for importing and exporting operations.

If you routinely download data from your company's mainframe, you'll find the Imp/Exp Setup... command invaluable. When you download the data, you're left with a file that contains the data in a certain data format—usually a fixed format in which the fields reside in the same position in every record. With the Imp/Exp Setup... command, you can create an Import/Export specification similar to the one shown in Figure A. An Import/Export specification tells Access where the fields reside in a file. When you import the data by using the Import... command, you can tell Access to use the Import/Export specification when retrieving the field values from the file.

Often, the external data files contain many fields, making the creation of Import/Export specifications a complex task. If you're like most users, you'd appreciate a hard-copy listing of long, complicated layouts. Unfortunately, Access doesn't offer a command that prints an Import/Export specification.

In this article, we'll show you how to access the layout information of all your Import/Export specifications. Armed with this information, you can create a query that compiles the data layout information and then you can create a report that puts the information on paper. At the end of the article, we'll show you how to apply the query and report to print Figure A's Import/Export specification.

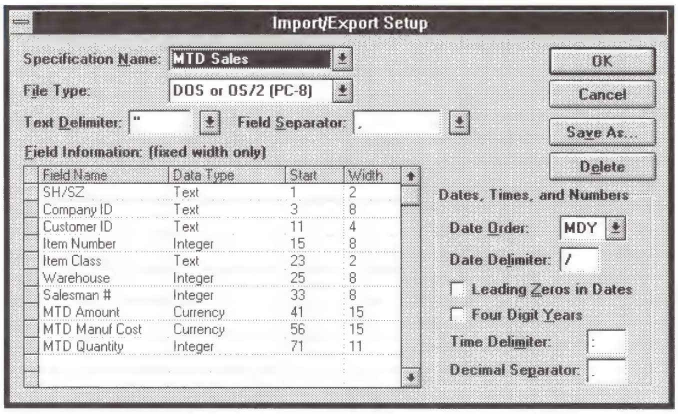
Opening Access' system tables

The key to printing the layout information is knowing where Access stores the Import/Export specifications. Access keeps all specifications you create in the MSysIMEXColumns and MSysIMEXSpecs system tables. You first create a query based on those tables to gather all the information you want on the report. You then create a report that prints the information.

Access maintains several system tables to hold a variety of information. Normally, you can't see the tables. However, you can view the system tables in your Database window by using the Options... command on the View menu. When you do, the Options dialog box will appear. Figure B shows a list of the General category's options you can set.

The second item in the General category is Show System Objects. When you set this option to *Yes* and then click OK, the system objects, such as the MSysIMEXColumns and MSysIMEXSpecs tables, will appear in the Database window, as shown in Figure C on page 10. Note that you'll have just read-only

Figure A



Specification Name: MTD Sales

File Type: DOS or OS/2 (PC-8)

Text Delimiter: "

Field Separator: ,

Field Information: (fixed width only)

Field Name	Data Type	Start	Width
SH/SZ	Text	1	2
Company ID	Text	3	8
Customer ID	Text	11	4
Item Number	Integer	15	8
Item Class	Text	23	2
Warehouse	Integer	25	8
Salesman #	Integer	33	8
MTD Amount	Currency	41	15
MTD Manuf Cost	Currency	56	15
MTD Quantity	Integer	71	11

Dates, Times, and Numbers

Date Order: MDY

Date Delimiter: /

□ Leading Zeros in Dates

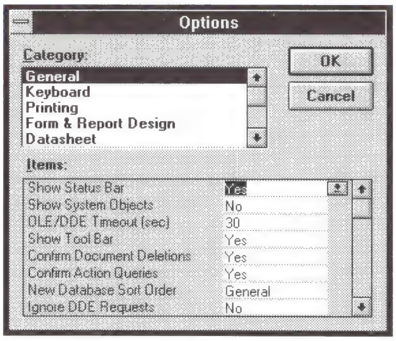
□ Four Digit Years

Time Delimiter: :

Decimal Separator: .

We'll show you how to create a report that prints the data layout of an Import/Export specification.

Figure B



Category:

General
Keyboard
Printing
Form & Report Design
Datasheet

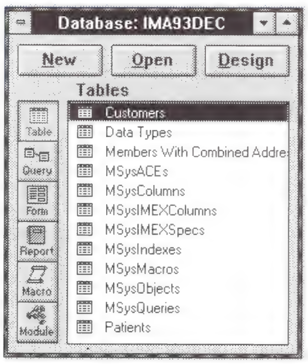
Items:

Show Status Bar	Yes
Show System Objects	No
OLE/DDE Timeout (sec)	30
Show Tool Bar	Yes
Confirm Document Deletions	Yes
Confirm Action Queries	Yes
New Database Sort Order	General
Ignore DDE Requests	No

You can set various characteristics of Access' operation in the Options dialog box.

access to the system tables. However, you don't need to change any data in these tables to create the Import/Export-specification report. You simply want to select information from the system tables.

Figure C



Database: IMA93DEC

- Customers
- Data Types
- Members With Combined Addre
- MSysACEs
- MSysColumns
- MSysIMEXColumns
- MSysIMEXSpecs
- MSysIndexes
- MSysMacros
- MSysObjects
- MSysQueries
- Patients

When you set the Show System Objects option to Yes, *your Database window will contain the system tables and other objects.*

To hide the system tables after you've finished working with them, you reset the Show System Objects option to *No*. You might wonder whether your query will be able to access the system tables after you remove them from view. Fortunately, you can. The tables remain online for your query's use as well as for Access' use.

The MSysIMEXColumns table

Most of the information that we need to include in our report resides in the table MSysIMEXColumns. Figure D shows the MSysIMEXColumns table after creating the Import/Export specification shown in Figure A.

Each row in this table corresponds to a field in the data layout. However, the table doesn't necessarily store the items in the same order as the Import/Export Setup dialog box shows.

Figure D

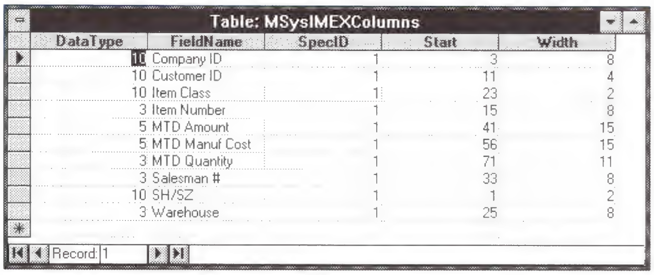


Table: MSysIMEXColumns

DataType	FieldName	SpecID	Start	Width
10	Company ID	1	3	8
10	Customer ID	1	11	4
10	Item Class	1	23	2
3	Item Number	1	15	8
5	MTD Amount	1	41	15
5	MTD Manuf Cost	1	56	15
3	MTD Quantity	1	71	11
3	Salesman #	1	33	8
10	SH/SZ	1	1	2
3	Warehouse	1	25	8

Record: 1

The rows of the MSysIMEXColumns table define the fields of the external data file's record.

As you can see, the table's first field stores the external field's data types. The second field includes the external table's field name. The third field stores the Import/Export specification's ID value (which Access assigns automatically). The last two fields hold the starting positions and widths of the external table's fields.

Notice that the DataType field stores a numeric value. These numbers correspond to the familiar Access data types you assign to fields as you create a table. The query we'll show you later includes this information to display the data type name of each field in the external file's record.

Note that Access doesn't include a system table that matches the data-type identifiers with the data type names. To include the data type names in a query, *you* must create the Date Types table, shown in Figure E. This table lists all the Access data types and the numbers that Access uses to identify them in the MSysIMEXColumns table.

Figure E

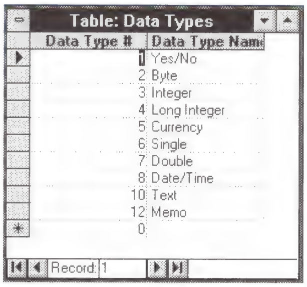


Table: Data Types

Data Type #	Data Type Nam
1	Yes/No
2	Byte
3	Integer
4	Long Integer
5	Currency
6	Single
7	Double
8	Date/Time
10	Text
12	Memo
0	

Record: 1

The Date Types table stores the data type names with their associated ID numbers.

The MSysIMEXSpecs table

While MSysIMEXColumns stores information about the individual fields of the external data file's layout, the MSysIMEXSpecs table contains general information about entire Import/Export specifications. Figure F shows the MSysIMEXSpecs table after you create the specification shown in Figure A.

We won't detail the information in this table. We're interested only in the SpecName field, which holds the name you assigned the Import/Export specification when you created it. You include this table in the query so you can print the specification name on the report.

Building the query

Now let's pull together all the information we've described. First, you must reveal the system tables in the database window. If you haven't already done so, pull down the View menu, select Options, and set Show System Objects to *Yes*. When you close the dialog box by clicking OK, the Database window will list the system tables.

Figure F

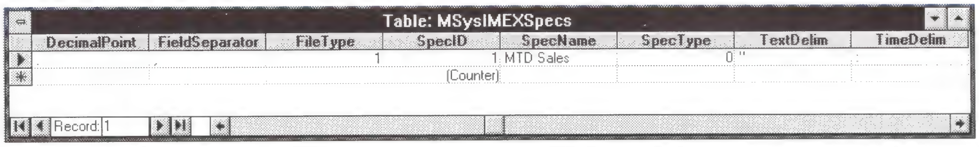


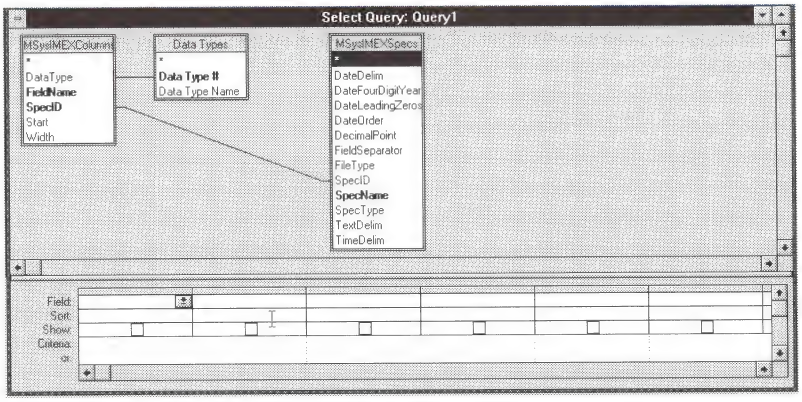
Table: MSysIMEXSpecs

	DecimalPoint	FieldSeparator	FileType	SpecID	SpecName	SpecType	TextDelim	TimeDelim
▶	.	,	1	1	MTD Sales	0	"	:
✱				(Counter)				

Record: 1

The MSysIMEXSpecs table holds information about entire Import/Export specifications.

Figure G



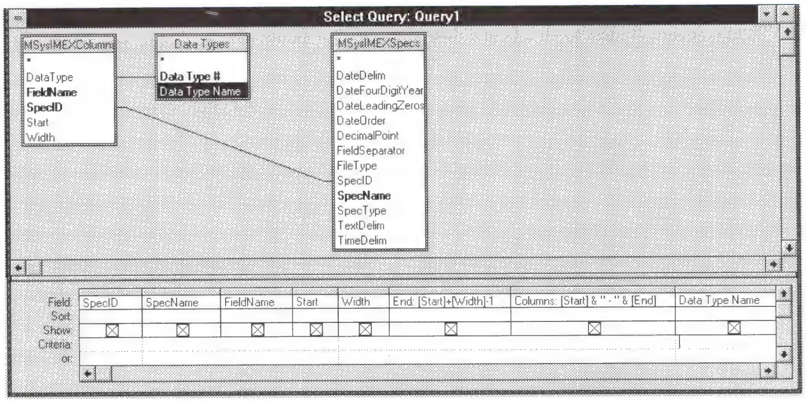
After you include the tables in the Query window and create the joins, your query will look like the one shown here.

In order to create the query, highlight the MSysIMEXColumns table in the Database window and click the New Query button (▣) on the tool bar. Next, press [F11] to return to the Database window and drag the MSysIMEXSpecs and Data Types tables to the new query.

After you include the three tables in the query, create the joins that link the tables. As you probably know, you create a join between two fields by dragging a field in one table's field list to the matching field in the other table's field list. For this query, you first have to drag the DataType field in the MSysIMEXColumns table's field list to the Data Type # field of the Data Types table. Next, you must drag the SpecID field of MSysIMEXColumns to the SpecID field of MSysIMEXSpecs. Your query should look like the one shown in Figure G. You may want to arrange and size the field lists of the three tables as shown in the figure. You'll be able to see the table relationships more easily.

Now fill out the QBE grid as shown in Figure H by performing the steps we list below. Each step describes how to create a column of the QBE grid.

Figure H



Field:	SpecID	SpecName	FieldName	Start	Width	End: [Start]+[Width]-1	Columns: [Start] & " - " & [End]	Data Type Name
Sort:								
Show:	☒	☒	☒	☒	☒	☒	☒	☒
Criteria:								
or:								

The steps we describe create this query.

- Drag the SpecID field from the MSysIMEXSpecs table to the first column's Field cell.
- Drag the SpecName field from the MSysIMEXSpecs table to the second column's Field cell.
- Drag the FieldName field from the MSysIMEXColumns table to the third column's Field cell.
- Drag the Start field from the MSysIMEXColumns table to the fourth column's Field cell.
- Drag the Width field from the MSysIMEXColumns table to the fifth column's Field cell.
- Enter the expression *End: [Start] + [Width]-1* in the sixth column's Field cell.
- Enter the expression *Columns: [Start] & "-" & [End]* in the seventh column's Field cell.
- Drag the Data Type Name field from the Data Types table to the eighth column's Field cell.

Finally, save the query by pulling down the File menu, selecting Save As..., and entering *Import Definitions* in the Save As dialog box.

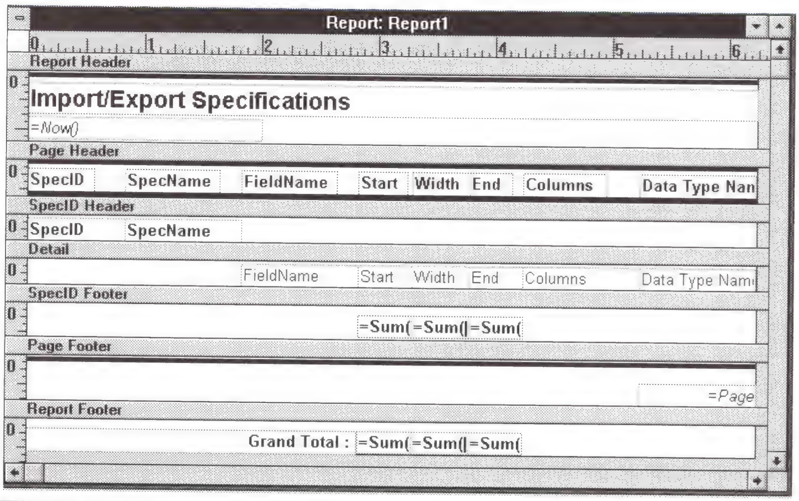
Creating the report

Now we'll create the report that prints the Import/Export specifications. Return to the Database window by pressing [F11] and highlight the Import Definitions query. Then, click the New Report button (▣) on the tool bar. In the New Report dialog box, click the ReportWizards button. In the next dialog box, select *Group/Totals* and click OK. Respond to the report wizard's dialog boxes as indicated in Table B. In the last dialog box, select the Fit All Fields On One Page check box and click the Design button. Figure I shows the report the report wizard generates.

Table B

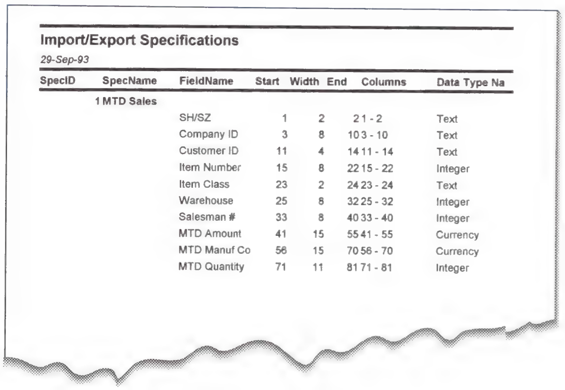
Report Wizard Question	Response/Selection
Which fields do you want on your report?	All fields
Which fields do you want to group by?	SpecID
How do you want to group data in each field?	Normal
Which fields do you want to sort by?	Start
What kind of look do you want for your report?	Presentation
What title do you want for your report?	Import/Export Specifications

Figure I



With the help of the Groups/Tools report wizard, you can create this report.

Figure J



Import/Export Specifications

29-Sep-93

SpecID	SpecName	FieldName	Start	Width	End	Columns	Data Type Na
1	MTD Sales						
		SH/SZ	1	2	2	1 - 2	Text
		Company ID	3	8	10	3 - 10	Text
		Customer ID	11	4	14	11 - 14	Text
		Item Number	15	8	22	15 - 22	Integer
		Item Class	23	2	24	23 - 24	Text
		Warehouse	25	8	32	25 - 32	Integer
		Salesman #	33	8	40	33 - 40	Integer
		MTD Amount	41	15	55	41 - 55	Currency
		MTD Manuf Co	56	15	70	56 - 70	Currency
		MTD Quantity	71	11	81	71 - 81	Integer

The Import Definitions report prints the field layout defined by the Import/Export specification named MTD Sales.

You'll need to clean up a few aspects of this default report. First, delete the text box controls that sum the numeric fields in the query. You'll find three such text boxes in the SpecID Footer section and another three in the Report Footer section. As you know, you delete a control in a report by first selecting the control and then pressing the [Del] key. In the Report Footer, you'll also need to delete the label control that displays the text *Grand Total:*.

When you've removed those controls, save the report by pulling down the File menu and selecting the Save As... command. Type *Import Definitions* in the Save As dialog box and click OK.

Creating the Import/Export specification

Now we'll show you how to create the Import/Export specification shown in Figure A. You can then run the Import Definitions report to create a printout of the external file's data layout.

Start by pulling down the File menu and selecting the Imp/Exp Setup... command. When the Import/Export Setup dialog box appears, fill out the Field Information grid as shown in Figure A. When you've finished, click the Save As... button. Then, enter *MTD Sales* in the Save As dialog box and click OK. Finally, click the OK button to close the Import/Export Setup dialog box.

Running the report

Now that you've entered an Import/Export specification, you can print the Import Definitions report. First, return to the Database window by pressing [F11] and then click the Report button. Highlight the Import Definitions report and click the Print Preview button (▣) on the tool bar. Then, click the Print... button on the tool bar and click OK in the Print dialog box. The report shown in Figure J will appear. ◆

A problem with parsing an Address field to derive Street Number and Street Name fields

The August issue of *Inside Microsoft Access* includes "Parsing an Address Field to Derive Street Number and Street Name Fields." I was able to follow your instructions for creating the query, and for almost all cases, the query worked fine. However, as I experimented with the technique, I found one type of address the query doesn't parse properly.

To test the query, I added to the sample table the address

```
6530 39th Avenue NE
```

and then ran the query. Instead of extracting the street number 6530, the query returned 653039. Essentially, the query ignores the space between the street number and the street name, *39th Avenue*.

The problem arises for all numbered streets. Apparently, the Val( ) function ignores the space between the street number and street name as it extracts the street number. Is there a way to tell the Val( ) function to stop when it encounters a space—just as the Val( ) function stops when it encounters other nonnumeric characters?

Loren F. Kahle
Austin, Texas

Mr. Kahle was the first reader to call our attention to this problem. He correctly points out that Val( ) ignores spaces in converting a string value to a numeric value. Before we show you how to work around this problem, we'll first review how the Val( ) function works and how we use the function to extract street numbers.

The Val( ) function

As you may know, Val( ) converts a string of numeric characters to numeric values. However, if the string contains nonnumeric characters, the function simply converts the characters in the string up to the first *nonnumeric* character. For instance, the function call

```
=Val("6530 Elm Street")
```

returns the value 6530.

Unfortunately, the Val( ) function doesn't operate quite as you'd expect. As Mr. Kahle reports, the Val( ) function ignores spaces when determining how many characters it includes in the number. This quirk causes problems when the address is for a street with a numeric name. In that situation, the function includes the street's numeric name along with the actual street number.

Parsing Address fields—a first try

Let's first review the table and query we used in the original article's example. Then, we'll show you how to correct the query. Figure A shows the Members With Combined Addresses table, which stores the addresses the query will parse.

Figure A

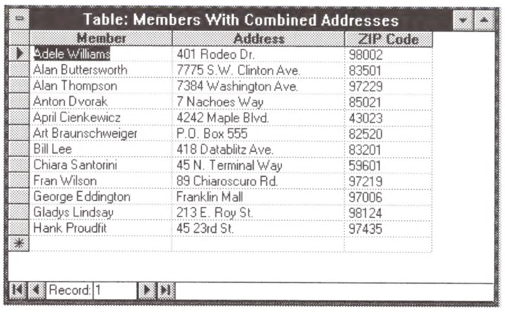


Table: Members With Combined Addresses

Member	Address	ZIP Code
Adele Williams	401 Rodeo Dr.	98002
Alan Buttersworth	7775 S.W. Clinton Ave.	83501
Alan Thompson	7384 Washington Ave.	97229
Anton Dvorak	7 Nachoes Way	85021
April Cienkewicz	4242 Maple Blvd.	43023
Art Braunschweiger	P.O. Box 555	82520
Bill Lee	418 Datablitz Ave.	83201
Chiara Santorini	45 N. Terminal Way	59601
Fran Wilson	89 Chiaroscuro Rd.	97219
George Eddington	Franklin Mall	97006
Gladys Lindsay	213 E. Roy St.	98124
Hank Proudfit	45 23rd St.	97435

We'll use this table to experiment with our address-parsing query.

If you created this table when working through the August article, you should note that we modified the last record of the table so that the address reads

```
45 23rd Street
```

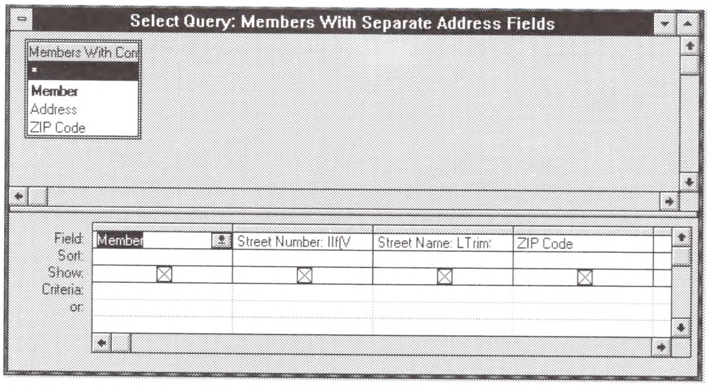
When you run the August article's query after making this change, you'll see that the query extracts the number 4523 as the Street Number entry.

Let's now quickly review that query. Figure B on page 14 shows the Members With Separate Address Fields query that parses the Address field into separate Street Number and Street Name fields. Unfortunately, the two expressions that define the Street Number and Street Name fields are too large to fit

Table A

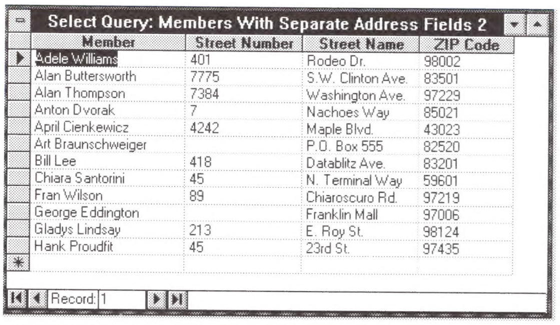
Column	Expression
2	Street Number: IIf(Val([Address]) = 0, "", Val([Address]))
3	Street Name: LTrim$(Right$([Address],Len([Address])-Len([Street Number])))

Figure B



The Members With Separate Address Fields query parses the Address field—but has a flaw.

Figure C



Select Query: Members With Separate Address Fields 2

Member	Street Number	Street Name	ZIP Code
Adele Williams	401	Rodeo Dr.	98002
Alan Buttersworth	7775	S.W. Clinton Ave.	83501
Alan Thompson	7384	Washington Ave.	97229
Anton Dvorak	7	Nachoes Way	85021
April Cienkewicz	4242	Maple Blvd.	43023
Art Braunschweiger		P.O. Box 555	82520
Bill Lee	418	Datablitz Ave.	83201
Chiara Santorini	45	N. Terminal Way	59601
Fran Wilson	89	Chiaroscuro Rd.	97219
George Eddington		Franklin Mall	97006
Gladys Lindsay	213	E. Roy St.	98124
Hank Proudfit	45	23rd St.	97435

The modified query returns the correct street number for those streets that have numeric names.

in the figure. Refer to Table A for the complete expressions.

As you can see, the Street Number field's expression uses the Val( ) function to strip off the leading numeric characters of the address fields. Of course, these numeric characters are the street number. The IIf( ) function tests whether the Val( ) function returns 0, which will happen when the Address field entry doesn't begin with a number.

Working around the problem

To use the Val( ) function to parse an address for a street with a numeric name, you must prevent Val( ) from ever seeing the characters beyond the first space. To do so, you must manipulate the Address field entry in such a way that the Val( ) function considers only the street number portion of the address.

You do so with the Left$( ) and InStr( ) functions. The Left$( ) function extracts a given number of characters from the left side of a string. The InStr( ) function searches a string for a given substring and returns the substring's starting position. We'll use InStr( ) to find the first space character in the Address entry and then use the result to tell Left$( ) how many characters to extract. That way, Left$( ) will return the characters up to the first space—in other words, the street number. Then, the Val( ) function can convert that result into a numeric value.

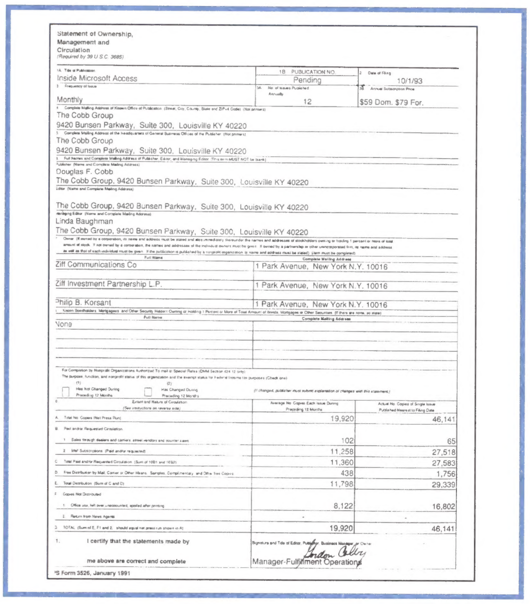
Now let's put these functions to work. Change the expression that defines the query's Street Number field from

```
Street Number: IIf(Val([Address]) = 0, "",
➥ Val([Address]))
```

to

```
Street Number: IIf(Val([Address]) = 0, "",
➥ Val(Left$([Address], InStr([Address]," "))))
```

When you run the query after updating the Street Number field's expression, the query will return the data shown in Figure C. ◆



Statement of Ownership, Management and Circulation
(Required by 39 U.S.C. 3685)

1A. Title of Publication Inside Microsoft Access	1B. PUBLICATION NO. Pending	2. Date of Filing 10/1/93
3. Frequency of Issue Monthly	3A. No. of Issues Published Annually 12	3B. Annual Subscription Price $59 Dom. $79 For.

4. Complete Mailing Address of Known Office of Publication (Street, City, County, State and ZIP+4 Code) (Not printers)
The Cobb Group
9420 Bunsen Parkway, Suite 300, Louisville KY 40220

5. Complete Mailing Address of the Headquarters of General Business Offices of the Publisher (Not printers)
The Cobb Group
9420 Bunsen Parkway, Suite 300, Louisville KY 40220

6. Full Names and Complete Mailing Address of Publisher, Editor, and Managing Editor (This item MUST NOT be blank)
Publisher (Name and Complete Mailing Address)
Douglas F. Cobb
The Cobb Group, 9420 Bunsen Parkway, Suite 300, Louisville KY 40220
Editor (Name and Complete Mailing Address)
The Cobb Group, 9420 Bunsen Parkway, Suite 300, Louisville KY 40220
Managing Editor (Name and Complete Mailing Address)
Linda Baughman
The Cobb Group, 9420 Bunsen Parkway, Suite 300, Louisville KY 40220

7. Owner (If owned by a corporation, its name and address must be stated and also immediately thereunder the names and addresses of stockholders owning or holding 1 percent or more of total amount of stock. If not owned by a corporation, the names and addresses of the individual owners must be given. If owned by a partnership or other unincorporated firm, its name and address as well as that of each individual must be given. If the publication is published by a nonprofit organization, its name and address must be stated.) (Item must be completed.)

Full Name	Complete Mailing Address
Ziff Communications Co	1 Park Avenue, New York N.Y. 10016
Ziff Investment Partnership L.P.	1 Park Avenue, New York N.Y. 10016
Philip B. Korsant	1 Park Avenue, New York N.Y. 10016

8. Known Bondholders, Mortgagees, and Other Security Holders Owning or Holding 1 Percent or More of Total Amount of Bonds, Mortgages or Other Securities (If there are none, so state)

Full Name	Complete Mailing Address
None	

9. For Completion by Nonprofit Organizations Authorized To mail at Special Rates (DMM Section 424.12 only)
The purpose, function, and nonprofit status of this organization and the exempt status for Federal income tax purposes (Check one)
(1) □ Has Not Changed During Preceding 12 Months (2) □ Has Changed During Preceding 12 Months (If changed, publisher must submit explanation of changes with this statement.)

10. Extent and Nature of Circulation (See instructions on reverse side)	Average No. Copies Each Issue During Preceding 12 Months	Actual No. Copies of Single Issue Published Nearest to Filing Date
A. Total No. Copies (Net Press Run)	19,920	46,141
B. Paid and/or Requested Circulation		
1. Sales through dealers and carriers, street vendors and counter sales	102	65
2. Mail Subscriptions (Paid and/or requested)	11,258	27,518
C. Total Paid and/or Requested Circulation (Sum of 10B1 and 10B2)	11,360	27,583
D. Free Distribution by Mail, Carrier or Other Means, Samples, Complimentary, and Other free Copies	438	1,756
E. Total Distribution (Sum of C and D)	11,798	29,339
F. Copies Not Distributed		
1. Office use, left over, unaccounted, spoiled after printing	8,122	16,802
2. Return from News Agents	-	-
G. TOTAL (Sum of E, F1 and 2 - should equal net press run shown in A)	19,920	46,141

11. I certify that the statements made by me above are correct and complete

Signature and Title of Editor, Publisher, Business Manager, or Owner
Gordon Colby
Manager-Fulfillment Operations

PS Form 3526, January 1991

Problems with 3-of-9 bar codes

I read with interest "Creating Bar Code Labels in Access Reports," in the September issue. I was pleased at your interest in bar coding, since I believe bar coding is a powerful productivity tool. Unfortunately, the article left out quite a bit of information a user needs to ensure success. I downloaded the CODE39.ZIP file from CompuServe and followed your instructions. As I suspected, the instructions produced bar code symbols that my bar code decoder couldn't read. I'll list the problems I encountered.

Providing the Start/Stop character

The USS-39 specification for Code 39 requires that you provide the Start/Stop character at the beginning and end of your encoded entry. Without this character, the decoder won't recognize the symbol as a legitimate bar code. The documentation accompanying the CODE39.ZIP file was minimal, but it did indicate that the @ character serves as the Start/Stop character in the RSCode39 font. (The choice of @ is a little confusing, since the industry standard is the * character.)

For example, suppose you want to encode the Product ID field entry *LK1871* with this font. You'd first create a text box control for printing the Product ID entries in the format @*LK1871*@. You set up the text box by assigning to its Control Source property the expression

```
="@¦[Product ID]¦@"
```

Next, you'd assign the RSCode39 font to the text box's Font Name property.

Bar code width and spacing

Another problem I encountered concerned the bar code font itself. Even after I included the Start/Stop characters, the report still produced unreadable bar codes. Examining the printout revealed problems with the bars'

Laying out forms and reports quickly by using default controls

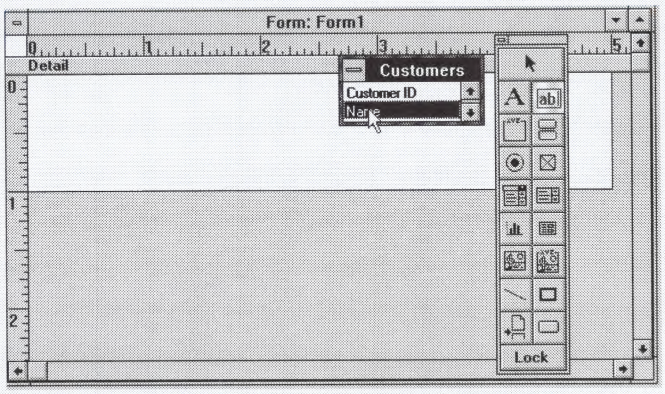
When you first began creating forms and reports, you probably learned to use both the tool box and the field list to place the controls. To place a control, you first select the Control tool from the tool box. You then drag the field you want from the list to the form or report. For example, Figure A shows how the screen looks when you're creating a text box control for the field called Name. Note that we first selected the Text Box tool in the tool box.

However, in many situations, this method wastes effort. You usually don't need to select a tool before placing the control. You can simply drag the field from the list onto the form or report. Why? Well, every data type has a default control type. Access will create that default control when you don't explicitly select a tool from the tool box.

The default control for almost all data types is the text box. However, the default properties of those controls may vary from data type to data type. For instance, the text box Access creates for a Memo field has a vertical scroll bar. The only data type whose default control is *not* a text box is the OLE Object data type. When you drag an OLE Object field to a form or report, Access will create a bound object frame for displaying the field value.

We don't mean to suggest that you don't need the tool box to design forms and reports. Many of its tools create controls that aren't default controls of a data type. For example, you may want to create a combo box for a field.

Figure A



Here, we're creating a text box control by using the tool box's Text Box tool.

Microsoft Access
Technical Support
(206) 635-7050

widths. First, you must understand that Code 39 is a two-width code. In other words, the bars and the spaces between the bars can be one of two possible widths. The ratio between the wide and narrow widths must be within 2.2:1 and 3:1. However, some widths were as large as 4:1 on my LaserJet. As a result, my scanner couldn't interpret them properly.

This is a common problem when using bar code fonts on printers that allow the font to be scaled. I don't know what printers the font RSCode39 was designed for, but you must be careful to specify the allowable font sizes for a particular printer. For example, those font sizes allowable for a 9-pin dot matrix printer are different from those for a laser printer.

Furthermore, I suspect there's another problem with the actual widths of the narrow-sized bar codes. When you use the 24-point font size for the text box (as you recommend in the original article), the narrow bars produced by the RSCode39 font are only 6.6 mils (0.0066"). This width is less than the recommended minimum for most bar code scanners. The hardware may not be able to read the bars.

What works

Although I had no luck with the RSCode39 font, I've successfully used TrueType bar codes in Windows applications—including Access. I recommend the fonts sold by Aedex Corporation in Placentia, California. You can reach the company at (714) 632-7000. While these fonts aren't free, they come with a very good manual that tells everything you need to know to produce bar codes your decoder can interpret. In addition, Aedex sells a very good, low-cost wand reader that connects to a serial port and places scanned data in the keyboard buffer.

Ernest E. Campbell
Sandy, Utah

We thank Mr. Campbell for sharing his extensive experience with TrueType bar code fonts. He's shared a number of practical suggestions you'll need to consider as you incorporate bar codes into your Access database.

Much of the same advice is appropriate for implementing the Postnet bar codes we described in the October issue of *Inside Microsoft Access*. In that article("Encoding ZIP+4 and Delivery Point Codes in the Postnet System"), we did a better job of describing the nuts and bolts of using the font. For instance, we described how to create the Start/Stop characters in which you must enclose the ZIP code entries. (Note that you call Start/Stop characters the frame bar characters when you're creating Postnet bar codes.)

However, you should carefully study the bar code symbols that the PostnetBars font creates. The US Postal Service has strict requirements for the width and height of the bars as well as their position on the envelope. You should contact your local post office and ask them to send you their booklet that describes the Postnet bar-coding specifications. ◆

Subscribe to Inside Microsoft Access Resource Disk!

Do you wish that you could experiment with the forms, reports, tables, macros, modules, and queries we regularly feature in *Inside Microsoft Access* but don't have the time or patience to create them? If so, you may want to subscribe to Inside Microsoft Access Resource Disk. Once you subscribe, we'll send you a disk loaded with all the useful tips featured in that month's *Inside Microsoft Access*. See the articles marked with the disk icon.

A six-month subscription to Inside Microsoft Access Resource Disk costs $29. A full one-year subscription is $49. If you don't want to subscribe but would like the forms, reports, tables, macros, modules, and queries in a particular issue of *Inside Microsoft Access*, you can purchase a single disk for only $9.95.

To subscribe or order a specific month's disk, just call Customer Relations at (800) 223-8720. Outside the US, please call (502) 491-1900.

Printed in the USA
Printed on recycled and recyclable paper
50% recycled fiber including 10% post-consumer waste